EL472378739

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Multi-Level Skimming of Multimedia Content Using Playlists

Inventor(s):
Anoop Gupta
Nosakhare D. Omoigui
Li-wei He

ATTORNEY'S DOCKET NO. MS1-279US

TECHNICAL FIELD

This invention relates to networked client/server systems and to methods of streaming and rendering multimedia content in such systems. More particularly, the invention relates to generating, maintaining and providing multiple skimmed versions of multimedia content using playlists.

BACKGROUND OF THE INVENTION

Multimedia streaming—the continuous delivery of synchronized media data like video, audio, text, and animation—is a critical link in the digital multimedia revolution. Today, streamed media is primarily about video and audio, but a richer, broader digital media era is emerging with a profound and growing impact on the Internet and digital broadcasting.

Synchronized media means multiple media objects that share a common timeline. Video and audio are examples of synchronized media—each is a separate data stream with its own data structure, but the two data streams are played back in synchronization with each other. Virtually any media type can have a timeline. For example, an image object can change like an animated .gif file, text can change and move, and animation and digital effects can happen over time. This concept of synchronizing multiple media types is gaining greater meaning and currency with the emergence of more sophisticated media composition frameworks implied by MPEG-4, Dynamic HTML, and other media playback environments.

The term "streaming" is used to indicate that the data representing the various media types is provided over a network to a client computer on a real-time, as-needed basis, rather than being pre-delivered in its entirety before playback. Thus, the client computer renders streaming data as it is received from a network server, rather than waiting for an entire "file" to be delivered.

In comparison to text-based or paper-based presentations, multimedia presentations can be very advantageous. Synchronized audio/visual presentations, for example, are able to capture and convey many subtle factors that are not perceivable from paper-based documents. Even when the content is a spoken presentation, an audio/visual recording captures gestures, facial expressions, and various speech nuances that cannot be discerned from text or even from still photographs.

Although streaming multimedia content compares favorably with textual content in most regards, one disadvantage is that it requires significant time for viewing. It cannot be "skimmed" like textual content. Thus, a "summarized" or "skimmed" version of the multimedia content would be very helpful.

Various technologies are available for "summarizing" or "previewing" different types of media content. For example, technology is available for removing pauses from spoken audio content. Audio content can also be summarized with algorithms that detect "important" parts of the content as identified by pitch emphasis. Similarly, techniques are available for removing redundant or otherwise "unimportant" portions or frames of video content. Similar schemes can be used with other types of media streams, such as animation streams and script streams.

Although such previewing techniques are available, these techniques typically require a significant amount of processing power to be performed and a significant amount of time to be completed. Such constraints make it difficult to generate previews "on the fly" as the data is being streamed to its destination.

One solution is to pre-generate and store a "preview" version of the multimedia content, thereby reducing the impact of "on the fly" calculations. However, generating and storing such a preview version creates a storage problem. The multimedia content itself frequently requires a significant amount of storage space. By storing an additional preview version of the multimedia content, the storage space requirements are increased further, thereby generating significant constraints on the media storage device. This problem is exacerbated if multiple preview versions are generated and stored.

The invention described below addresses these disadvantages of previewing multimedia content, providing an improved way to generate and maintain such preview content.

SUMMARY OF THE INVENTION

A system includes a multimedia server computer that can provide multimedia content, as well as skimmed versions of the multimedia content, to one or more client computers. A skimmed version of the multimedia content is a preview or summary of the multimedia content that can be presented to a user in less time than presenting the entire multimedia content would require.

One or more skimmed versions of multimedia content are provided by the server computer using playlists. Skimming information is maintained by the server computer for each skimmed version, the skimming information identifying particular segments of the multimedia content for a particular skimmed version. The server computer (or alternatively the client computer) uses the skimming information to generate a playlist of multimedia segments of the multimedia content. Rather than maintaining the actual segments of the multimedia content, the playlist identifies segments of the multimedia content. The playlist is used by the server computer to access the appropriate segments of the multimedia content and provide such segments to the client computer(s).

Additionally, a user can select different skimmed versions that he or she will be presented with. The user can make such selections prior to or during presentation of a skimmed version of the multimedia content. Upon selecting a different skimmed version, one of the server computer or the client computer generates a playlist for the newly selected skimmed version and determines a location in the new playlist that corresponds to the location being presented in the current playlist. Presentation of the new skimmed version then begins at the corresponding location in the new playlist.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings. The same numbers are used throughout the figures to reference like components and/or features.

Fig. 1 shows a client/server network system and environment in accordance with the invention

Fig. 2 shows a general example of a computer that can be used as a server or client in accordance with the invention.

Fig. 3 is an exemplary block diagram showing the generation of skimming level information for multimedia content.

Fig. 4 illustrates a multimedia file of Fig. 3 in more detail.

Fig. 5 is a flowchart illustrating an exemplary process for generating skimming level information in accordance with the invention.

Fig. 6 illustrates exemplary client and server computers in which the playlist for the skimmed version is generated at the server computer.

Fig. 7 illustrates alternate client and server computers in which the playlist for the skimmed version is generated at the client computer.

Fig. 8 is a flowchart illustrating exemplary steps in presenting multimedia segments corresponding to a skimming level to a user in accordance with the invention.

Fig. 9 is a flowchart illustrating exemplary steps in changing skimming levels in accordance with the invention.

Fig. 10 shows one implementation of a graphical user interface window that displays multimedia content at a client computer.

Fig. 11 shows another implementation of a graphical user interface window that displays multimedia content at a client computer.

DETAILED DESCRIPTION

General Network Structure

Fig. 1 shows a client/server network system and environment in accordance with the invention. Generally, the system includes one or more network server computers 102, and multiple (*n*) network client computers 104. The computers communicate with each other over a data communications network. The communications network in Fig. 1 comprises a public network 106 such as the Internet. The data communications network might also include local-area networks and private wide-area networks.

Multimedia server 102 has access to streaming media content in the form of different media streams. These media streams can be individual media streams (e.g., audio, video, graphical, etc.), or alternatively composite media streams including multiple such individual streams. Some media streams might be stored as files 108 in a database or other file storage system, while other media streams 110 might be supplied to the server on a "live" basis from other data source components through dedicated communications channels or through the Internet itself.

Multimedia server 102 also has access to data or information identifying different skimmed versions of the media streams. This "skimming" information identifies different segments of media streams that are part of a particular skimmed version of that stream. Multiple skimming versions or "skimming levels" can be maintained for each media stream. By using the skimming information to identify portions of media streams, storage space requirements are reduced because the data of the media streams need not be duplicated.

In the discussions to follow, the multimedia content available to the client computers 104 is discussed as being streaming media. However, it should be noted that the invention can also be used with "pre-delivered" media rather than streaming media, such as media previously stored at the client computers 104 via the network 106, via removable magnetic or optical disks, etc.

Streaming Media

In this discussion, the term "composite media stream" describes synchronized streaming data that represents a segment of multimedia content. The composite media stream has a timeline that establishes the speed at which the content is rendered. The composite media stream can be rendered to produce a plurality of different types of user-perceivable media, including synchronized audio or sound, video graphics or motion pictures, animation, textual content, command script sequences, or other media types that convey time-varying information or content in a way that can be sensed and perceived by a human. A composite media stream comprises a plurality of individual media streams representing the multimedia content. Each of the individual media streams corresponds to and represents a different media type and each of the media streams can be rendered by a network client to produce a user-perceivable presentation using a particular presentation medium. The individual media streams have their own timelines, which are synchronized with each other so that the media streams can be rendered simultaneously for a coordinated multimedia presentation. The individual timelines define the timeline of the composite stream.

There are various standards for streaming media content and composite media streams. "Advanced Streaming Format" (ASF) is an example of such a standard, including both accepted versions of the standard and proposed standards for future adoption. ASF specifies the way in which multimedia content is stored, streamed, and presented by the tools, servers, and clients of various multimedia vendors. ASF provides benefits such as local and network playback, extensible media types, component download, scalable media types, prioritization of streams, multiple language support, environment independence, rich inter-stream relationships, and expandability. Further details about ASF are available from Microsoft Corporation of Redmond, Washington.

Regardless of the streaming format used, an individual data stream contains a sequence of digital data sets or units that are rendered individually, in sequence, to produce an image, sound, or some other stimuli that is perceived by a human to be continuously varying. For example, an audio data stream comprises a sequence of sample values that are converted to a pitch and volume to produce continuously varying sound. A video data stream comprises a sequence of digitally-specified graphics frames that are rendered in sequence to produce a moving picture.

Typically, the individual data units of a composite media stream are interleaved in a single sequence of data packets. Various types of data compression might be used within a particular data format to reduce communications bandwidth requirements.

The sequential data units (such as audio sample values or video frames) are associated with both delivery times and presentation times, relative to an arbitrary start time. The delivery time of a data unit indicates when the data unit should be delivered to a rendering client. The presentation time indicates when the value should be actually rendered. Normally, the delivery time of a data unit precedes its presentation time.

The presentation times determine the actual speed of playback. For data streams representing actual events or performances, the presentation times correspond to the relative times at which the data samples were actually recorded. The presentation times of the various different individual data streams are consistent with each other so that the streams remain coordinated and synchronized during playback.

Exemplary Computer Environment

In the discussion below, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by one or more conventional personal computers. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. In a distributed computer environment, program modules may be located in both local and remote memory storage devices.

Fig. 2 shows a general example of a computer 130 that can be used as a server or client in accordance with the invention. Computer 130 is shown as an example of a computer that can perform the functions of a server computer 102 or a client computer 104 of Fig. 1.

Computer 130 includes one or more processors or processing units 132, a system memory 134, and a bus 136 that couples various system components including the system memory 134 to processors 132.

The bus 136 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 138 and random access memory (RAM) 140. A basic input/output system (BIOS) 142, containing the basic routines that help to transfer information between elements within computer 130, such as during start-up, is stored in ROM 138. Computer 130 further includes a hard disk drive 144 for reading from and writing to a hard disk, not shown, a magnetic disk drive 146 for reading from and writing to a removable magnetic disk 148, and an optical disk drive 150 for reading from or writing to a removable optical disk 152 such as a CD ROM or other optical media. The hard disk drive 144, magnetic disk drive 146, and optical disk drive 150 are connected to the bus 136 by an SCSI interface 154 or some other appropriate interface. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for computer 130. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 148 and a removable optical disk 152, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs) read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 148, optical disk 152, ROM 138, or RAM 140, including an operating system 158, one or more application programs 160, other program modules 162, and program data 164. A user may enter commands and information into computer 130 through input devices such as keyboard 166 and pointing device 168. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to the processing unit 132 through an interface 170 that is coupled to the bus 136. A monitor 172 or other type of display device is also connected to the bus 136 via an interface, such as a video adapter 174. In addition to the monitor, personal computers typically include other peripheral output devices (not shown) such as speakers and printers.

Computer 130 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 176. The remote computer 176 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 130, although only a memory storage device 178 has been illustrated in Fig. 2. The logical connections depicted in Fig. 2 include a local area network (LAN) 180 and a wide area network (WAN) 182. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet. In the described embodiment of the invention, remote computer 176 executes an Internet Web browser program such as the "Internet Explorer" Web browser manufactured and distributed by Microsoft Corporation of Redmond, Washington.

When used in a LAN networking environment, computer 130 is connected to the local network 180 through a network interface or adapter 184. When used in a WAN networking environment, computer 130 typically includes a modem 186 or other means for establishing communications over the wide area network 182, such as the Internet. The modem 186, which may be internal or external, is connected to the bus 136 via a serial port interface 156. In a networked environment, program modules depicted relative to the personal computer 130, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Generally, the data processors of computer 130 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems are typically distributed, for example, on floppy disks or CD-ROMs. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable storage media when such media contain instructions or programs for implementing the steps described below in conjunction with a microprocessor or other data processor. The invention also includes the computer itself when programmed according to the methods and techniques described below. Furthermore, certain sub-components of the computer may be programmed to perform the functions and steps described below. The invention includes such sub-components when they are programmed as described. In addition, the invention described herein includes data structures, described below, as embodied on various types of memory media.

For purposes of illustration, programs and other executable program components such as the operating system are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computer, and are executed by the data processor(s) of the computer.

Generating Skimmed Versions

Multiple preview or skimmed versions of multimedia content can be created, such versions being referred to as being different "skimming levels". Each of these different skimming levels provides a different level of detail of the multimedia content, and thus typically includes a different total presentation time. For example, a first skimming level may represent little of the original multimedia content and have a presentation time of 15 minutes rather than the 2 hour presentation time of the entire multimedia content. A second skimming level may represent more of the original multimedia content and have a presentation time of 1 hour.

Fig. 3 is an exemplary block diagram showing the generation of skimming level information for multimedia content. Multimedia content 200 is received by a skimming generator 202. Skimming generator 202 can be implemented in hardware or software, such as a software program executing on a computer 130 of Fig. 2. Additionally, skimming generator 202 can be implemented in server computer 102 of Fig. 1, or alternatively in another computer (not shown) either coupled to or independent of network 106. Multimedia content 200 can be provided to skimming generator 202 in a variety of different manners, such as streaming of a live presentation, streaming of a data file, "pre-delivery" of a data file (e.g., on a CD-ROM or transferred via network 106 of Fig. 1), etc.

Skimming generator 202 processes the multimedia content 200 to create multiple (*m*) skimming levels 204, 206, and 208 corresponding to the multimedia content 200. Skimming generator 202 separates the multimedia content into multiple segments and generates the multiple skimming levels 204 – 208 using various combinations of these segments. Each of the skimming levels 204 – 208 comprises a different set of these multimedia segments. Skimming generator 202 uses any of a variety of conventional summarizing or previewing technologies (e.g., pitch analysis to detect important parts of audio content and similar techniques to identify important parts of video content) to generate the skimming levels 204 – 208.

The results of the various previewing techniques for different streams may identify different portions of the multimedia content that are more important. In the illustrated example, this situation is resolved by using a composite scoring method to identify which segments are more important (and thus are kept as part of the skimmed version), and which segments are less important (and thus are not included as part of the skimmed version).

Alternatively, the results of one of the previewing techniques on a single data stream may be used to identify which segments are to be dropped. For example, a single data stream (e.g., the audio stream) may be evaluated, with the results of that evaluation being used to identify which segments of the audio stream (and corresponding segments of the video and other streams) are dropped without any evaluation of the corresponding segments of the other streams.

Skimming information for each of the skimming levels 204 – 208 is then stored in multimedia file 210. This skimming information can be, for example, identifies of particular segments of the multimedia content, importance rankings for each of multiple segments of the multimedia content, etc. Additionally, an indication of the total number *m* of skimming levels is also stored in multimedia file 210.

In the illustrated example, the multimedia content 200 is received by skimming generator 202 as multimedia file 210. Thus, skimming generator 202 stores the skimming information for each of the skimming levels 204 – 208 back into the same data file as the multimedia content 200 is stored in.

In the example illustrated in Fig. 3, multimedia file 210 is an ASF file. Multimedia file 210 includes a header portion 212 and a data portion 214. Header portion 212 contains data representing various control and identifying information regarding the multimedia file 210. Data portion 214 contains the multimedia content as well as the skimming information for each of the skimming levels 204 – 208.

The skimming level generation process identifies different segments of the multimedia content 200 for each of the different levels 204 – 208. Skimming generator 202 then stores data identifying these different segments in the data portion 214 of multimedia file 210. Alternatively, a linear separation technique could be used to delineate the segments, such as each segment being a 5-second portion of the multimedia content.

Alternatively, rather than storing identifiers of particular segments of the multimedia content, skimming generator 202 could generate particular "rankings" for each segment of the multimedia content. These rankings are generated using the conventional summarizing or previewing technologies to identify which portions of the multimedia content are more important than which others or alternatively could be generated manually. The different portions are then assigned a particular rank or weight (e.g., "high", "medium", and "low"; or any of an infinite number of rankings (such as real number values between zero and one)). These rankings can then be subsequently used to dynamically identify which segments should be presented for a particular skimming level.

Additionally, skimming generator 202 identifies the relationship between the presentation timeline of the original multimedia content and the segments identified by the skimming information. This relationship may be stored as an additional stream in multimedia file 210, or alternatively as one or more index tables associated with multimedia file 210. The relationship is a mapping of presentation times of the skimmed version to the original multimedia content, indicating for any presentation time of the skimmed version, what the corresponding presentation time of the original multimedia content is. For example, the data 35 seconds into the skimmed version may correspond to 120 seconds into the original multimedia content. This stored relationship thus allows the server (or client) computer, during subsequent playback of a skimmed version, to identify the current presentation point with respect to the original multimedia content. A similar mapping is maintained for presentation times of the original multimedia content to locations of the skimmed version (e.g., presentation times, byte offsets into the skimmed stream, segment identifiers, etc.).

Fig. 4 illustrates a multimedia file 210 in more detail. Multimedia file 210 includes header portion 212 containing various control and identifying information regarding the multimedia file 210. Header portion 212 includes data identifying each of the streams in data portion 214, and optionally may include the number of different skimmed versions maintained in data portion 214. Skimming information for each skimmed version is maintained as a stream in data portion 214, referred to as a "skimming stream".

Data portion 214 includes data representing multiple (x) streams 220, 222, 224, 226, and 228. Streams 220 – 228 include media stream data for the multimedia content, such as audio data and video data of a composite media stream, as well as skimming streams that include skimming information for the multimedia content.

In the illustrated example, streams 224 and 226 are skimming streams that include "markers" (e.g., time ranges) used to identify the segments of the multimedia content. The markers can be used to generate a "playlist" identifying particular segments of the multimedia content that are to be provided for the corresponding skimming level. A playlist includes a reference to the multimedia content, as well as start and end times for one or more segments of the multimedia content. Alternatively, a skimming stream may include rankings or weights for each of multiple segments of the multimedia content.

In the illustrated playlists of Fig. 4, the segments are identified by start and end times corresponding to the timeline of the original multimedia content. Thus, the playlist 230 identified by stream 224 indicates the first five seconds (0-5) of the multimedia content, as well as the seventh through ninth seconds (7-9), seventeenth through twenty-second seconds (17-22), thirty-seventh through forty-sixth seconds (37-46), fifty-second through sixty-first seconds (52-61), and seventy-second through seventy-seventh seconds (72-77) of the multimedia content. Similarly, the playlist 232 identified by stream 226 indicates the first four seconds (0-4) of the multimedia content, as well as the twenty-second through twenty-seventh seconds (22-27), thirty-second through thirty-ninth seconds (32-39), and fifty-second through fifty-seventh seconds (52-57) of the multimedia content.

Fig. 5 is a flowchart illustrating an exemplary process for generating skimming level information in accordance with the invention. The process of Fig. 5 is implemented by skimming generator 202 of Fig. 3, and may be performed in software. Fig. 5 is described with additional reference to components in Figs. 3 and 4.

Initially, multimedia content is received by skimming generator 202 (step 250). Skimming generator 202 then determines which segments of the multimedia content correspond to a skimming level (step 252). As discussed above, the generation of different segments can be accomplished using any of a variety of conventional previewing techniques.

Skimming generator 202 also stores skimming information identifying the segments determined in step 252 as a stream of multimedia file 210 corresponding to the multimedia content (step 254). Skimming generator 202 then checks whether additional skimming levels are to be generated (step 256). The number of skimming levels and their level of detail can be pre-programmed into skimming generator 202, or alternatively can be manually input by a user.

Skimmed Version Presentation

When providing a skimmed version of the multimedia content to a user, server computer 102 of Fig. 1 accesses multimedia file 210 of Fig. 4 for the stream 220 – 228 corresponding to the requested skimming level. Server computer 102 then generates a playlist for that stream that identifies which of the segments of the multimedia content are to be provided to the client as the skimmed version. Alternatively, the client computer could generate the playlist.

Fig. 6 illustrates exemplary client and server computers in which the playlist for the skimmed version is generated at the server computer. Client computer 104 includes a multimedia player 280 that provides a user interface (UI) allowing a user to be presented with streaming multimedia content. The multimedia player 280 may be incorporated into the operating system or run as a separate, self-contained application. In either case, the multimedia player operates in a graphical user interface windowing environment such as provided by the "WINDOWS" brand of operating systems, available from Microsoft Corporation of Redmond, Washington.

Multimedia player 280 communicates with a multimedia presentation module 282 of server computer 102. Multimedia presentation module 282 streams the multimedia content to multimedia player 280 for presentation to the user. Multimedia presentation module 282 can stream the entire multimedia content to multimedia player 280 for presentation. Additionally, multimedia presentation module 282 can distinguish between streams of multimedia content and streams that contain skimming information. Multimedia presentation module 282 uses the skimming information to transmit a skimmed version of the multimedia content to the multimedia player 280 as well.

Multimedia presentation module 282 includes a skimming module 284 and a location identifier module 286. Skimming module 284 controls the provision of skimming level options to the user, allowing the user to select (via the interface of multimedia player 280) a skimmed version for presentation. Additionally, skimming module 284 also provides multimedia presentation module 282 with the control to access skimming information and provide the segment(s) of the multimedia content corresponding to the skimming information to client computer 104.

In the illustrated example, skimming module 284 accesses the skimming information (e.g., in multimedia file 210 of Fig. 4) corresponding to a user-selected skimming level. Skimming module 284 uses this information to generate a playlist for the skimming level. Multimedia presentation module 282 uses the playlist generated by skimming module 284 to identify which segments of the multimedia content to provide to the client computer 104 as the selected skimmed version of the multimedia content. Alternatively, rather than comprising skimming information from which a playlist is generated, the stream in multimedia file 210 could comprise a playlist that can be accessed by skimming module 284 "as is", without requiring any additional generating step.

Alternatively, in situations where the skimming information is a rank for each segment of the multimedia content, skimming module 284 uses the rankings to generate an appropriate playlist. Skimming module 284 uses a user-selected skimming level as a threshold for generating the playlist. For example, skimming module 284 includes in the playlist any segments having a ranking equal to or greater than the threshold.

A user, through the interface provided by multimedia player 280, is able to select different skimmed versions by selecting a different skimming level. This selection can occur prior to being presented with a skimmed version and/or while being presented with a skimmed version.

When a user changes the skimming level, multimedia player 280 provides, to multimedia presentation module 282, information identifying the current presentation time of the multimedia segment being provided to the user. This current time information could be a reference to the original multimedia content (e.g., 36 minutes and 20 seconds into the original multimedia content), or alternatively an identification of the current segment of the skimmed version being presented and an offset into that segment (e.g., five seconds into the third segment of the skimmed version).

Location identifier module 286 uses the information provided by multimedia player 280 (either current presentation time or current segment and offset) to determine a new location in the playlist of the newly selected skimming level. As discussed above, a mapping of each skimmed version to the original multimedia presentation is part of (or stored separately but corresponding to) the multimedia file 210 that includes the skimming information. Using these mappings, location identifier module 286 is able to identify the location in the new skimmed version to which the current location of the current skimmed version corresponds.

Location identifier module 286 identifies the location in the new playlist by accessing the mapping for the current skimmed version using the current location in the current skimmed version. The mapping (e.g., an index table) identifies a location in the original multimedia content that corresponds to the current location in the current skimmed version. The identified location from the original multimedia content is then used to access the mapping for the new skimmed version, which identifies a location in the new skimmed version that corresponds to the identified location of the original multimedia content, and thus to the current location in the current skimmed version.

Alternatively, additional mappings can be maintained that alleviate the necessity for such a "two-step" lookup process. Direct skimmed version to skimmed version mappings can be generated and maintained (either by server 102 or by skimming generator 202 of Fig. 3) that map locations in one skimmed version to corresponding locations of other skimmed versions.

Fig. 7 illustrates alternate client and server computers in which the playlist for the skimmed version is generated at the client computer. Client computer 104 includes a multimedia player 280 that provides an interface for the user to be presented with streaming multimedia content. Multimedia player 280 communicates with a multimedia presentation module 288 of server computer 102. Multimedia presentation module 288 streams the multimedia content to multimedia player 280 for presentation to the user. Multimedia presentation module 288 can stream the entire multimedia content to multimedia player 280 for presentation, or alternatively a skimmed version(s) of the multimedia content.

Multimedia presentation module 288 includes a skimming module 290 that controls the provision of skimming level options to the user. Skimming module 290 allows the user to select (via the interface of multimedia player 280), a skimmed version for presentation. Skimming module 290 also provides the skimming information corresponding to the multimedia content to playlist generator 292 of client 104. Multimedia player 280 communicates a user-selection of a skimming level to playlist generator 292, which in turn uses the skimming information to generate a playlist for the skimming level. This generated playlist is transferred to multimedia presentation module 288 of server 102, which in turn uses the generated playlist to identify which segments of the multimedia content to provide to the client computer 104 as the selected skimming version of the multimedia content.

Additionally, a user is able, through the interface provided by multimedia player 280, to change the skimmed version he or she is being presented with. The user can select an initial skimming level and/or change the current skimming level while being presented with a skimmed version. When a user changes the skimming level, location identifier module 294 determines the proper location within the playlist of the newly selected skimming level.

When a user changes the skimming level, multimedia player 280 provides, to location identifier module 294, information identifying the current presentation time of the multimedia segment being provided to the user. Location identifier module 294 uses this information to determine a new location in the playlist of the newly selected skimming level in a manner analogous to location identifier module 286 of Fig. 6.

Fig. 8 is a flowchart illustrating exemplary steps in presenting multimedia segments corresponding to a skimming level to a user in accordance with the invention. The steps on the left side of Fig. 8 are implemented by client computer 104 of Fig. 6, and the steps on the right side of Fig. 8 are implemented by server computer 102. The steps of Fig. 8, on both client and server computers, may be performed in software. Fig. 8 is described with additional reference to components in Fig. 6.

Initially, the client computer 104 receives a user request for multimedia content (step 302). The request can be initiated by the user in any of a variety of conventional manners, such as selection of a multimedia title in a graphical user interface (GUI), a menu selection, a command-line input, etc. Client computer 104 communicates the user request to server computer 102 (step 304), such as by sending a message to server computer 102.

Server computer 102, upon receipt of the request, accesses the multimedia file corresponding to the request and provides the skimming level information regarding the multimedia content to client computer 104 (step 306). Client computer 104 presents the skimming level information to the user (step 308). Based on the presented information, the user can select one of the skimming levels. Client computer 104 receives the skimming level selection (step 310) and communicates the selection to server computer 102 (step 312).

Server computer 102, upon receipt of the skimming level selection, accesses the skimming information and generates the playlist for the selected skimming level (step 314). Alternatively, the playlist could be generated by client computer 104 as discussed above with reference to Fig. 7. Server computer 102 then provides the segments of the multimedia content that are identified by the playlist generated in step 314 to client computer 104 (step 316). These segments are received by client computer 104, which in turn presents the segments to the user (step 318).

Various optimizations may also be implemented to improve the quality of the presentation of the multimedia content when streaming the segments of the multimedia content identified by a playlist to client computer 104. One such optimization is pre-buffering of the multimedia content at client computer 104. Subsequent segments of multimedia content can be buffered at client computer 104 while current segments are being presented to the user. Thus, client computer 104 can seamlessly switch from presentation of the current segments to presentation of the next segments in the playlist.

Additionally, multimedia content may be streamed as multiple frames, including independent frames and dependent frames. Independent frames include all of the information necessary to present (e.g., display video or play audio) a frame (or sample) of data, while dependent frames identify only differences between the dependent frame and one or more previous frames (either dependent or independent). Playlists may include segments that begin at either independent frames or dependent frames. If the beginning of a segment is at a dependent frame, then additional information prior to the beginning of that segment is needed in order to generate the appropriate data for the dependent frame.

This situation can be resolved in a variety of different manners. In one implementation, the additional information (e.g., the previous independent frame and possibly intervening dependent frames) is transmitted from server computer

102 to client computer 104. This can result in a noticeable pause to the user while the additional information is processed. In another implementation, if the beginning points for segments are known in advance, additional "specialized" independent frames can be generated as necessary in advance that include the necessary additional information. In this implementation, the specialized independent frame is transmitted to client computer 104 along with the first dependent frame of the segment, thereby alleviating client computer 104 from having to process additional information spread over potentially numerous independent and dependent frames.

Fig. 9 is a flowchart illustrating exemplary steps in changing skimming levels in accordance with the invention. The steps of Fig. 9 are implemented by server computer 102, and may be performed in software. Alternatively, steps 332 – 338 could be implemented by client computer 104. Fig. 9 is described with additional reference to components in Fig. 6.

Initially, server computer 102 receives an indication of a new skimming level request (step 332). Upon receipt of the indication, server computer 102 generates a playlist for the newly selected skimming level (step 334). Server computer 102 then identifies the current location in the current playlist that is being presented to the user (step 336). Using this current location, server computer 102 determines the corresponding location in the playlist for the new skimming level (step 338). Server computer 102 then determines the start location within the new playlist (step 340). Server computer then provides the segments of the multimedia content identified by the new playlist to the client computer beginning at the start location (step 342).

In the illustrated embodiment, the start location within the new playlist determined in step 340 is the beginning of the segment corresponding to the location identified in step 338. For example, if the user requests a new skimming level at a presentation time that corresponds to five seconds into the seventh segment of the new playlist, then the start location is determined in step 340 to be the beginning of the seventh segment of the new playlist. Alternatively, the start location could be determined in step 340 to be five seconds into the seventh segment of the new playlist.

User Experience

Fig. 10 shows one implementation of a graphical user interface window 352 that displays multimedia content at a client computer 104 of Fig. 1. The user interface 352 is provided by multimedia player 280 of Fig. 6 or Fig. 7. The UI window 352 includes a video screen 354, a graphics screen 356, and a text screen 358.

Video screen 354 is the region of the UI within which the video portion of the multimedia content is rendered. If the multimedia content does not include video data, screen 354 displays static or dynamic images representing the content. For audio content, for example, a dynamically changing frequency wave that represents an audio signal can be displayed in screen 354.

Graphics screen 356 is the region of the UI within which the graphics portion of the multimedia content is rendered. The graphics portion can include, for example, a set of slides or presentation foils that correspond to the video portion. If the multimedia content does not include graphics data, then the graphics screen 356 is left blank (or an indication given that no graphics are available).

Text screen 358 is the region of the UI within which the text portion of the multimedia content is rendered. The text portion can include, for example, a table of contents that outlines the multimedia content. If the multimedia content does not include text data, then the text screen 358 is left blank (or an indication given that no graphics are available).

The UI window 352 also includes a command bar 360, shuttle controls 362, a volume control 364, summary level selectors 366, 368, and 370, and content information space 372. Command bar 360 lists familiar UI commands, such as "File", "View", and so forth.

Shuttle controls 362 allow the user to control playback of the multimedia content. Shuttle controls 362 include a stop button, a pause button, rewind buttons, a play button, and fast forward buttons. Selection of the fast forward (or rewind buttons) cause the multimedia player to jump ahead or back in the media presentation by a predetermined amount (e.g., one second, five seconds, to the next segment, etc.). The play, stop, and pause buttons cause their conventional functions to be performed by media player 280.

Three different summary buttons 366, 368, and 370 are included corresponding to different summary levels. Selection of summary button 366 causes multimedia player 280 to present a skimmed version of the multimedia content having a first level of detail to the user. Similarly, selection of summary button 368 causes multimedia player 280 to present a skimmed version of the multimedia content having a second level of detail, while selection of summary button 370 causes multimedia player 280 to present a skimmed version of the multimedia content having a third level of detail.

The user can actuate one of the summary buttons 366-370 via a UI actuation mechanism, such as a pointer or by tabbing to the desired play button and hitting the "enter" key. Upon selection of a summary button, the multimedia player presents the skimmed version of the multimedia content corresponding to the selected skimming level.

Similarly, the user can actuate any of the buttons of the shuttle controls 362 via a UI actuation mechanism, such as a pointer or by tabbing to the desired play button and hitting the "enter" key. Upon selection of a button, the multimedia player performs the requested action (e.g., stops or pauses playback, rewinds, etc.).

Volume control 364 allows the user to adjust the volume of the audio portion of the multimedia content.

Content information space 372 lists information pertaining to the multimedia content being rendered on the screens 354 – 358. The content information space includes the show name, author and copyright information, and tracking/timing data.

Fig. 11 shows another implementation of a graphical user interface window that displays multimedia content at a client computer 104 of Fig. 1. The user interface 382 is provided by multimedia player 280 of Fig. 6 or Fig. 7.

Many of the components of UI window 382 are analogous to those of UI window 352 of Fig. 10. Like UI window 352 of Fig. 10, UI window 382 includes a video screen 384, a graphics screen 386, a text screen 388, a command bar 390, shuttle controls 392, a volume control 394, and content information space 396. Each of these is analogous to the corresponding components of UI 352 of Fig. 10.

UI 382 also has a menu 398 associated with skimming button 400. In this illustration, menu 398 is a drop-down or pull-down menu that opens beneath skimming button 400 in response to actuation of a tab 402. Alternatively, menu 398 may be invoked by placing a pointer over skimming button 400 and right clicking a mouse button.

Menu 398 lists multiple skimming levels from which a user can select. In the illustrated example, five skimming levels are listed: level 1 (15 minute presentation duration), level 2 (30 minute presentation duration), level 3 (45 minute presentation duration), level 4 (1 hour presentation duration), and level 5 (1 ½ hours presentation time). The user can select one of the listed skimming levels to instruct the multimedia player to present the corresponding preview content. The user can select a new skimming level after the multimedia player has begun presentation by invoking the menu and selecting the new level. In response, the multimedia player presents a new skimmed version corresponding to the new skimming level.

Figs. 10 and 11 are merely exemplary illustrations of user interfaces via which a user can select a skimming level. Alternatively, other interfaces could be used via which the user can change the skimming level, such as a rotatable dial, a sliding scale, an alphanumeric input control (e.g., allowing the user to type in a number, letter, or word), etc.

Conclusion

The invention provides multi-level skimming of multimedia content using playlist. A playlist for a skimmed version of the multimedia content is generated from skimming information maintained along with the multimedia content. The skimming information advantageously identifies segments of the multimedia content, thereby conserving storage space by eliminating the need to duplicate storage of the actual segments.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.